

**BCS Higher Education Qualification**  
**Profession Graduate Diploma**  
**March 2019**  
**EXAMINERS' REPORT**  
**Programming Paradigms**

**General comments<sup>1</sup>**

The overall picture for this module is much the same as in previous years. As reported in previous years, questions that cover topics including logic programming, concurrent systems and functional programming, are often not answered as well, although there were some good answers. This continues to illustrate that students would benefit from expanding the range of material studied in preparation for the course, looking to cover the full range of the syllabus.

For example, functional programming is an area that candidates should be familiar with. There are more functional languages available, e.g. Haskell and F#, and other programming languages are including aspects of functional programming. It is even more relevant to learn and understand this programming paradigm. Logic programming provides another different style of developing software and something that gives a broader appreciation of the ways that you can solve problems.

Candidates and centres are encouraged to look at this topic in more detail and where possible try out some of the basic concepts as part of studying for this module. The syllabus is also being updated in the summer of 2019 and it will include some additional suggestions for learning resources.

**Question number: A1**

**Syllabus area:**

Nature of Programming Languages 1.6 Language Standardisation

**Total marks allocated: 25**

**Examiners' Guidance Notes**

Answers to part a) were generally good when talking about imperative programming styles but weren't always able to write much about declarative styles. Some answers were able to provide examples of statements, but other answers would have benefited from some examples. Answers to part b) weren't as detailed and were typically quite general. Better answers were able to talk about more issues, such as those in the answer pointers.

**Question number: A2**

**Syllabus area:**

Programming Environments 2.4 Debugging Tools, 2.5 Testing Tools and  
2.2 Interactive Development Environments

**Total marks allocated: 25**

**Examiners' Guidance Notes**

Answers were of a good standard for both parts of the question. Candidates were able to differentiate between the terms and offer descriptions that demonstrated an understanding of the issues. It would be good to see some more detail in the answers to expand on the points raised.

**Question number: A3**

**Syllabus area:** Object Orientation 3.1

**Total marks allocated: 25**

**Examiners' Guidance Notes**

This was generally well answered. Better answers were able to go beyond the basic discussion of OO and show how the concepts are relevant to the software system for drawing diagrams. A number of answers used diagrams, but some were generic diagrams rather than ones that were related to the example in the question. Some answers about encapsulation focused more on information hiding (e.g. using Java's access modifiers) rather than the issue of bringing together data and operators on that data.

**Question number: B4**

**Syllabus area:**

Concurrency

**Total marks allocated: 25**

**Examiners' Guidance Notes**

There were some good answers to the question, showing an appreciation of issues in concurrency. A number of answers talked about concurrency in general terms rather than thinking about the issues for the specific scenario. Some candidates created an entirely different scenario that was not an online clothing shop, and then described that instead. When there is a scenario in the question, answers should refer to the scenario. For part (b) there were many answers that discussed Java solutions to concurrency issues (for example monitors), but the question specifically asked about the concurrency control for the database. Concurrency control features for databases should be discussed.

<b>Question number: B5</b>
<b>Syllabus area:</b> Logic Programming
<b>Total marks allocated: 25</b>
<b>Examiners' Guidance Notes</b>
Very few candidates attempted this question. There were some good answers. However, some answers demonstrated limited understanding of logic programming and other answers talked generally about logic programming but did not relate that to the problem set in the question. When learning about logic programming, it would be good to try some of the concepts within Prolog to help your understanding of the issues. Candidates should also take care to use correct technical language. For part (a), not enough candidates explained what implications the CWA has for the reasoning in Prolog. For part (b), many candidates incorrectly thought that both would be the same, and incorrectly thought that all possible values for X would be generated by food(X) before checking meat(X). Prolog has a depth first search order.